

04/27/00
jc796 U.S. PTO

05-01-00

IN THE UNITED STATES PATENT AND TRADEMARK OFFICE

Inventorship..... Allen et al.
Applicant..... Microsoft Corporation
Attorney's Docket No. MS1-487US
Title: Web Address Converter for Dynamic Web Pages

TRANSMITTAL LETTER AND CERTIFICATE OF MAILING

To: Commissioner of Patents and Trademarks
Washington, D.C. 20231

From: Kasey C. Christie (509) 324-9256
Lee & Hayes, PLLC
421 W. Riverside Avenue, Suite 500
Spokane, WA 99201

jc530 U.S. PTO
09/560703
04/27/00

The following enumerated items accompany this transmittal letter and are being submitted for the matter identified in the above caption.

1. Transmittal Letter with Certificate of Mailing included.
2. PTO Return Postcard Receipt
3. Check in the Amount of \$2,938.00
4. Fee Transmittal
5. New patent application (title page plus 48 pages, including claims 1-56 & Abstract)
6. Executed Declaration
7. 5 sheets of formal drawings (Figs. 1-5)
8. Assignment w/Recordation Cover Sheet

Large Entity Status [x]

Small Entity Status []

The Commissioner is hereby authorized to charge payment of fees or credit overpayments to Deposit Account No. 12-0769 in connection with any patent application filing fees under 37 CFR 1.16, and any processing fees under 37 CFR 1.17.

Date: 4/27/00

By:

Kasey C. Christie
Reg. No. 40,559

CERTIFICATE OF MAILING

I hereby certify that the items listed above as enclosed are being deposited with the U.S. Postal Service as either first class mail, or Express Mail if the blank for Express Mail No. is completed below, in an envelope addressed to The Commissioner of Patents and Trademarks, Washington, D.C. 20231, on the below-indicated date. Any Express Mail No. has also been marked on the listed items.

Express Mail No. (if applicable)

EL580804056

Date: 4.27.2000

By:

Lori A. Vierra

IN THE UNITED STATES PATENT AND TRADEMARK OFFICE

APPLICATION FOR LETTERS PATENT

Web Address Converter for Dynamic Web Pages

Inventor:

Joshua Allen

Jeff L. Ranck

ATTORNEY'S DOCKET NO. MS1-487US

002240 00209550

TECHNICAL FIELD

This invention relates to accessing dynamic World Wide Web ("Web") pages on a Web server by converting Universal Resource Locator (URL) addresses from one type to another.

BACKGROUND

There are more than a billion documents available on the World Wide Web ("Web") over the Internet and this number continues to rapidly increase. These documents ("Web pages") are stored as files on Web servers. Each of these Web pages has a unique Web address. These address are also called Uniform Resource Locators (URLs) or Universal Resource Locators (URLs). URLs are more fully explained in RFC 1738 "Uniform Resource Locators (URL) Berners-Lee, Masinter & McCahill." Static Web Pages and Static Addressing

An Internet device, such as a computer using a Web browser, typically accesses a specific Web page by providing its unique Web address (e.g., a URL). That Web page is a static file stored on a Web server. The file is simply copied without change to the requesting Internet device. Every device accessing the static file sees the same results. The stored file remains unchanged until an authorized user actively modifies the file. These types of Web pages are typically called "static." A typical URL for a static Web page looks like this:

http://domain.name.com/pagename.htm

The "http://" is the value of the scheme field and it identifies the protocol scheme being used to transmit over the Internet. For the Web, the protocol scheme

1 typically is HyperText Transfer Protocol (HTTP). The "domain.name.com" is the
2 value of the hostname field and it identifies the domain (or the Web server) that
3 hosts the Web page addressed by the static URL. The actual format of this field
4 depends upon the domain name conventions observed. Typically, the format
5 includes a domain name and an extension (e.g., microsoft.com).

6 The "pagename" is the value of the path field and/or the file-name field. It
7 may include a path to the specific Web page. It includes the file name of the
8 specific Web page. The ".htm" is the value of the file-extension field and it
9 identifies the format of the file. In this example, the format of the static file is the
10 most common format for a Web page: HyperText Markup Language (HTML).

11 12 Dynamic Web Pages and Dynamic Addressing

13 The opposite of a static Web page is a "dynamic" Web page. A dynamic
14 Web page is one that is created the moment the page is accessed and it is usually
15 created based upon data in a database. Unlike a static Web page, a dynamic Web
16 page that a viewer sees is not stored intact on a Web server. Instead, a dynamic
17 Web page is generated anew each time it is accessed.

18 A dynamic Web page is generated based upon a stored file containing
19 instructions and an associated database. Therefore, each instance of a generated
20 dynamic Web page may be different from a previously generated page using the
21 same address. There are many different implementations of dynamic Web pages.
22 The implementation differs from each other in the set of instructions used in the
23 stored file on the Web server and the type of database accessed. Examples of such
24 implementations include Active Server Pages (ASP) by the Microsoft Corporation
25 and "JavaBeans" Activation Framework (JAF).

1 A typical URL for a dynamic Web page may look like this:

2
3 *http://domain.name.com/pagename.asp?parm1=val1&parm2=val2*

4
5 This example uses an ASP implementation. The protocol scheme, hostname, path,
6 and filename fields are the same as those fields in the static URL. However, there
7 are fields in a dynamic address that are different from fields in a static address.

8 The extension “.asp” is a value of a file-extension field and identifies the
9 format of the dynamic-page-generation instructions. The extension “.asp”
10 indicates that the page is formatted as an Active Server Page (ASP). The “?”
11 symbol is a signal that the URL points to a dynamic page and it separates the
12 portion of the dynamic URL referring to a specific file and the portion of the URL
13 containing parameters.

14 The “parm1=” and “parm2=” elements identify the names of categorized
15 parameter. The values of these parameters are used to generate the dynamic Web
16 page. “val1” and “val2” are the values of the parameters. The values are typically
17 used to access items in a database. A parameter consists of a parameter name and
18 its associated value. There can be a series of many parameters. The “&” symbol
19 separates each parameter for the other parameters.

20 21 Web Search Engines and Spiders

22 No central bibliographic authority exists to catalog the information found
23 on the tens of millions of Web sites on the Internet. Generally, two basic
24 approaches are available for finding the proverbial needle in this immense Web
25 haystack: a subject directory or a search engine.

1 Subject directories, such as "Snap" and "MSN", catalog Web pages and
2 organize them by subject. Each Web page is manually (or automatically) analyzed
3 and categorized. Users can browse through the various categories and
4 subcategories in the subject directories to find a Web site on a particular topic.
5 Typically, Web pages are categorized and added to the directory by professional
6 Web searchers or by user submissions.

7 A search engine provides a searchable database of indexed keywords. A
8 search engine examines Web pages for specified keywords and returns a list of the
9 Web pages where the keywords were found. Although search engines are general
10 class of programs, the term is often used to specifically describe systems like "Alta
11 Vista" and "Excite" that enable users to search for Web pages on the Web.

12 A search engine includes two main parts: index searcher and index
13 generator. An index searcher includes a database of indexing keywords of Web
14 pages and logic for searching that database. An index generator includes a
15 "spider" for gathering Web pages and an "indexer" for generating an index into
16 those pages.

17 Typically, a search engine works by sending out the spider to fetch as many
18 pages as possible. The indexer then reads these pages and creates an index based
19 on the words contained in each page. Each search engine typically uses a
20 proprietary algorithm to create its indices such that, ideally, only meaningful
21 results are returned for each query.

22 Spiders are sometimes referred to as "Web-spiders", "robots", "Web
23 wanderers", "crawlers", "Web-crawler", "ants", or "worms." These alternative
24 names refer to programs that have the same basic functionality to visit Web sites
25 by requesting documents from them.

1 A spider will "crawl" a Web page by following links found on the page.
2 Normal Web browsers (e.g., "Internet Explorer") are not spiders, because they are
3 operated by humans, and don't automatically retrieve referenced documents.

4 Provided with a page by a spider, an indexer parses the document and
5 inserts selected keywords into the database with references back to the original
6 location of the source page. How this is accomplished depends on the indexer.
7 Some indexers index the titles of the Web pages or the first few paragraphs. Some
8 parse the entire contents and index all words. Some parse the meta-tag or other
9 special hidden tags.

10 Meta-tags are special HTML tags that provide information about a Web
11 page. Unlike normal HTML tags, meta-tags do not affect how the page is
12 displayed. Instead, they provide information such as who created the page, how
13 often it is updated, what the page is about, and which keywords represent the
14 page's content. Many search engines use this information when building their
15 indices.

16 When visiting a Web site, most spiders will check a file called the
17 "robots.txt" file. This file informs the spider whether the spider is authorized to
18 search the site and if so authorized, which pages on the site to retrieve.

19 Single-destination Web sites called "portals" are often a combination of a
20 "subject directory" and a "search engine." These portals include a search engine
21 (with its spider and indexer) or are closely associated with a third-party search
22 engine. These portals often include an organized and customized subject
23 directory.
24
25

1 The Invisible Web

2 The Invisible Web is made up of information stored in Web databases.
3 Unlike pages on the visible Web, information in databases is generally
4 inaccessible to the spiders to compile search engines.

5 Search engines typically index the Web by visiting Web pages and
6 indexing their content. In particular, the spiders use the links found on pages to
7 find new Web pages. The links include static URLs.

8 Most spiders tend to ignore the content of a dynamic Web address and thus,
9 the contents of the referenced dynamic Web page. These dynamic Web pages are
10 often ignored because the format of their dynamic URL is different from the URL
11 format of a static Web page. Spiders are often specifically programmed to ignore
12 dynamic addresses because of the complexity of navigating through dynamic
13 pages.

14 The information found in the databases of dynamic Web sites is not
15 indexed by search engines. Therefore, these dynamic Web sites are not found by
16 those using search engines to search the Web. This huge, unmapped region of the
17 Internet is called the "Invisible Web."

18 E-commerce sites with on-line shopping catalogs typically use dynamic
19 Web pages because their databased inventory is changing constantly. These sites
20 wish to be indexed by search engines because to help bring users to their site.
21
22
23
24
25

Conventional Solution

To allow search engines to index their sites, dynamic sites (such as e-commerce sites with inventory) periodically generate “snapshots” of their dynamic Web pages. These snapshots are static Web pages generated from corresponding dynamic Web pages, which are generated at a moment in time.

However, there are several significant drawbacks to the “snapshot” approach. In a short period of time, the snapshots no longer represent the current inventory. Periodically generating the snapshots consumes processing and storage resources.

Although the snapshot approach does allow a search engine to index the dynamic Web site, the URLs stored by the search engine are static URLs. Therefore, the search engine ultimately directs a user to the snapshot pages rather than to the preferable dynamic pages. Dynamic sites would prefer users to use their dynamic page to take full advantage of the dynamic nature of the site. If the users are using the snapshot pages, then the information seen by the user may not be accurate.

SUMMARY

A Web address converter converts dynamic Web pages in a manner that overcomes the drawbacks of the “snapshot” approach. The Web address converter helps dynamic Web sites get the attention of the spiders of Internet search engines. With the Web address converter, requests from Web browsers using static addresses access corresponding dynamic Web pages and requests from search engines generate an instance of a Web page having links with static addresses pointing to corresponding dynamic Web pages.

1 The Web address converter performs both Dynamic-to-Static (D-to-S)
2 address conversion and Static-to-Dynamic (S-to-D) address conversion. D-to-S
3 address conversion is done when generating a spider-friendly main page for a
4 spider of a search engine to crawl. S-to-D address conversion is used when a
5 browser uses a static address to access a corresponding dynamic Web page. The
6 static address that the browser uses was originally created when the spider-friendly
7 main page was generated.

8 9 **BRIEF DESCRIPTION OF THE DRAWINGS**

10 Fig. 1 is a schematic illustration of an exemplary computer network (such
11 as the Internet) that includes a Web site hosting dynamic Web pages. The site
12 includes a converter implementing an embodiment of the Web address converter
13 for dynamic Web pages.

14 Fig. 2 is a bi-directional address-conversion-mapping diagram showing:

- 15 • from top to bottom, an exemplary implementation of
- 16 dynamic-to-static (D-to-S) address conversion; and
- 17 • from bottom to top, an exemplary implementation of static-to-
- 18 dynamic (S-to-D) address conversion.

19 Fig. 3 is flowchart showing a process implementing the Web address
20 converter for dynamic Web pages.

21 Fig. 4 is flowchart showing a process implementing the Web address
22 converter for dynamic Web pages.

23 Fig. 5 is an example of a computer capable of implementing the Web
24 address converter for dynamic Web pages.

DETAILED DESCRIPTION

The following description sets forth a specific embodiment of the Web address converter for dynamic Web pages that incorporates elements recited in the appended claims. This embodiment is described with specificity in order to meet statutory written description, enablement, and best-mode requirements. However, the description itself is not intended to limit the scope of this patent. Rather, the inventor has contemplated that the claimed Web address converter might also be embodied in other ways, in conjunction with other present or future technologies.

The following description sets forth a Web address converter for dynamic Web pages that enable the use of static addresses to access corresponding dynamic Web pages. The converter provides static to dynamic URL mapping for certain incoming static addresses. The converter also provides for the dynamic generation of an instance of a Web page containing links to dynamic Web pages, but the links include static addresses pointing to the dynamic Web pages.

Dynamic Web Sites and the Internet

Fig. 1 shows a computer network that includes computers linked via Internet 20. Client 22 using a Web browser (e.g., "Internet Explorer" by the Microsoft Corporation) may access the Internet via one or more Internet Service Providers (ISPs) 24.

Fig. 1 shows a search engine (with a spider) 26 connected to the Internet 20. The search engine 26 includes a Web server (or collection of Web servers) 28 and a database 29. The database 29 stores indexed keywords associated with source Web pages.

1 The search engine 26 contains the searchable database 29 of keywords that
2 are associated with Uniform Resource Locators (URLs) pointing to Web pages.
3 Typically, a search engine uses a "spider" program module to fetch as many
4 documents (i.e., Web pages) as possible. An "indexer" of the search engine reads
5 these documents and creates a database based on the words contained in each
6 document.

7 Fig. 1 also shows a static Web site 30 connected to the Internet 20. The
8 static Web site 30 includes a Web server (or collection of Web servers) 32. The
9 static Web site 30 contains static Web pages stored in a file located in the file
10 system of the Web server 32, and stored in non-volatile memory, such as disk
11 drives 34a-34d.

12 Disk drives 34a-34d illustrate a hierarchical path to a file containing a static
13 Web page on drive 34d. Suppose that drive 34a is named "root"; drive 34b is
14 named "sub1"; drive 34c is named "sub2"; drive 34d is named "sub2"; and a file
15 36 named "file.htm". The path to file 36 would be "/root/sub1/sub2/sub3/file.htm"

16 Although the path is shown as multiple disk drives 34a-34d, the path
17 typically will be both file directories and subdirectories on the same file system of
18 the Web server. Alternatively, the path may be across multiple Web servers.

19 Fig. 1 shows a dynamic Web site 40 connected to the Internet 20. The
20 dynamic Web site 40 includes a Web server (or collection of Web servers) 42. The
21 dynamic Web pages are dynamically generated by the server 42 based upon data
22 stored in database 44.

23 The dynamic Web site 40 also includes a Web address converter 46. The
24 converter 46 implements the exemplary embodiment of the Web address
25 converter. The converter 46 may be a filter designed to examine incoming

Web Address Mapping

Fig. 2 shows an example of Web address mapping for converting Web addresses from one type to another type that is implemented by Web address converter 46. This address conversion is done so that requests from Web browsers using static addresses can access dynamic Web pages. Also, it is done so that requests from spiders of search engines generate an instance of a spider-friendly Web page having links with static addresses pointing to dynamic Web pages.

There are two types of Web address conversion: Static-to-dynamic (S-to-D) and dynamic-to-static (D-to-S). Each is the reverse of the other. S-to-D address conversion is performed to redirect requests from Web browsers using static addresses to corresponding dynamic Web pages. D-to-S address conversion is performed to generate an instance of a spider-friendly Web page having links with static addresses pointing to corresponding dynamic Web pages.

Fig. 2 shows the address conversion mapping generally at 100. Reading from top to bottom shows D-to-S address conversion as represented by “down” arrow 102. Reading from bottom to top shows S-to-D address conversion as represented by “up” arrow 104. Unless specifically stated otherwise, the addition of any field or value in one conversion direction corresponds to the removal of the same field or value in the other conversion direction, and vice versa.

Block 110 shows a generic example of a dynamic address at 112. The dynamic address points to a dynamic Web page. The dynamic address 112 shown in Fig. 2 is

`http://hostname/path.asp?parmx=valx&parmlast=vallast`

Block 120 shows the fields removed from the dynamic address 112 to convert the dynamic address of 112 to a static address *that points to the same dynamic Web page*. The following fields are removed:

- a field-separator symbol (“/”) at 122;
- a dynamic file extension (“.asp”) and a file-parameter-separator symbol (“?”) at 124;
- one or more parameter names (“parmx”) and a symbol (“=”) that equates each of them to value at 126;
- a file-parameter-separator symbol (“&”) used to separate each parameter from each other parameter at 128 (a parameter includes a parameter name and its value); and
- a last parameter name (“parmlast”) in the series of parameters and the symbol (“=”) that equates it to its value at 130.

If there is only one parameter, then “parmlast” will be the only parameter name and there will be no “&” symbol.

Block 140 shows the fields that are found in both the source Web address and the converted Web address. Generally, these common fields are the basis for converting from one address type to another. These common fields map from the original address to specific positions in the converted address.

The common fields include “scheme” field at 142 that identifies the protocol scheme being used to transmit the request for a Web page and the protocol scheme to be used to send the requested Web page. Since the exemplary embodiment is used for converting Web addresses, the typical protocol scheme is HTTP; therefore, the value of the scheme field 142 normally is “http://”.

Other common fields shown in block 140 include a "hostname" field at 144, a "path" at 146, a "valx" at 148, and a "vallast" at 150.

The hostname field 144 contains a name of a Web server (or Web site) hosting the dynamic Web page. The name in the hostname field at 144 is "hostname" and it is derived from the static Web address 112.

The path field 146 contains a name of a hierarchical path used to access the file containing instructions for generating the dynamic Web page. The name in the path field at 146 is "path" and is derived from the static Web address 112. Similar to the hierarchical path illustrated by the disk drives 34a-34d of Fig. 1, the path in the path field 146 may be one or more levels. For example, the path may be "path"; "root/path"; or "root/subpath1/subpath2/path".

The valx value field 148 may be one or more parameter values. Each valx value field contains a value associated with a parameter name (like "parmx" 126). The "=" symbol equates each parameter with a specific value. For example, "parmx" has a value of "valx".

The vallast value field 150 a value associated with the last parameter name (like "parmlast" 130). The "=" symbol equates the last parameter name with a specific value. For example, "parmlast" has a value of "vallast". If there is only one parameter in the dynamic address 112, then "vallast" will be the only value and there will be no valx values.

Block 160 shows the fields added to the common fields of block 140 to convert the dynamic address of 112 to a static address *that points to the same dynamic Web page*. The following fields are added:

- a field-separator symbol (“/”) and an alias-indicator flag (“flag”) at 162;
- one or more field-separator symbol (“/”) at 164;
- a static file extension (“.htm”) that indicates that the format of the file is static although the file’s format is actually dynamic.

The alias-indicator flag 162 is inserted into the resulting static address so that the exemplary converter will recognize it as a static address that does not point to an actual static Web page. Rather, the address with the alias-indicator flag points to a dynamic Web page.

Block 170 shows an example of a converted static address at 172. This static address *points to the same dynamic Web page* that the dynamic address of 112 does. The static address 172 shown in Fig. 2 is

`http://hostname/flag/path/valx/vallast.htm`

The arrows between the fields of block 160 and the static address 172 of block 170 illustrate the relative mapping of the fields into the static address. The fields 162-166 are inserted between the common fields 142-150 of block 140 to form the static address 172.

In Fig. 2, the common fields 142-150 of block 140 are shown underlined in the dynamic address 112 and the static address 172. The underlining makes the common fields easier to locate in the originating dynamic address 112 and the

1 resulting static address 172. The underlining also highlights the existence and the
2 relative location of each field in the both addresses. Furthermore, it distinguishes
3 the common fields from other fields that are specific to only one type of address.

4 The arrows between blocks 110 and 130 illustrate which fields are removed
5 from the dynamic address 112 and where the fields are removed. Likewise, the
6 arrows between blocks 160 and 170 illustrate which fields are added to the
7 common fields to form the static address 172 and where the fields are added.

8 The above description of Fig. 2 is given reading from top to bottom to
9 illustration D-to-S address conversion. However, as mentioned above, the
10 *addition* of any field or value for the above-described D-to-S address conversion
11 corresponds to the *removal* of the same field or value in the S-to-D address
12 conversion. Likewise, the *removal* of any field or value for the above-described
13 D-to-S address conversion corresponds to the *addition* of the same field or value
14 in the S-to-D address conversion.

15 16 Alternative Parameter Value Mapping

17 In the exemplary embodiment of the Web address converter, the converter
18 places the parameter values (such as “valx” or “vallast”) based upon their relative
19 position in the original dynamic address. For example, “valx” is before “vallast”.
20 When such a static address is converted back (re-mapped) to a dynamic address,
21 the converter assumes the parameter fields based upon the relative positioning of
22 the parameter values in the static address. For example, “parmlast” is assumed to
23 be equal to “vallast” because “vallast” is the last field in the static address.

1 In an alternative embodiment, the parameter associations may be specified
2 in the static address so that values are re-mapped back specific parameters. This
3 may occur in a variety of ways. As one example, assume the following original
4 dynamic address:

5
6 `http://hostname/path.asp?parmA=valA&parmB=valB&parmC=valC`

7
8 This dynamic address may be converted into this static address:

9
10 `http://hostname/flag/path/parmA=valA/parmB=valB/parmC=valC.htm`

11
12 An alternative encoding may be:

13 `http://hostname/flag/path/parmA/valA/parmB/valB/parmC/valC.htm`

14
15 Any encoding that includes both the parameter name and the value in such
16 a way that the original pairing can be extracted may be used. When such an
17 alternative embodiment of the Web address converter maps this static address
18 backs to its original dynamic address, it will not assume how to map the values
19 back to the dynamic address. Rather, it will examine the static address to see
20 exactly how to map the values to which parameters because the parameters
21 themselves are specified in the static address.

1 Method of Web Address Analysis and Conversion

2 Fig. 3 shows a process for handling a request for dynamic Web pages that
3 is implemented by the Web address converter. At 200, a request to access a Web
4 page is received by a Web server or a filter (such as the filter 46 in Fig. 1). For the
5 exemplary implementation, the filter receives the request. Such a request may be
6 a GET request in HTTP.

7 If the request includes a dynamic address, then the filter passes it back to
8 the Web server for normal dynamic Web page invocation. Otherwise, the request
9 includes a static Web address. At 202, the filter parses the request to extract the
10 static address included therein.

11 At 204, the static address is examined to determine if it includes an "alias-
12 indicator flag." The alias-indicator flag is any set of unique (within the Web
13 server) alphanumeric characters that may be used within a static address to
14 identify the address as being an "alias address." For example, the flag may be
15 "root", "flag", "static", or "alias". The field containing the alias-indicator flag is
16 shown at 162 in Fig. 2. An "alias address" is a static address that points to a
17 dynamic Web page rather than a static one. There are two sources of an "alias
18 address."

19 The first source is where an alias address is a resulting static address from
20 the exemplary D-to-S address conversion. During the D-to-S address conversion,
21 the alias-indicator flag is inserted into the resulting static address in the manner
22 shown at blocks 160 and 170 in Fig. 2.

23 The second source of an alias address is an address stored in a file on the
24 Web server. That file is called the "robots.txt" file. The address stored in this file
25 includes the alias-indicator flag.

002240" E0209550

1 This "robots.txt" file informs a spider whether the spider is authorized to
2 search the site and if so authorized, which pages on the site to retrieve. In the
3 exemplary embodiment, the "robots.txt" file authorizes a spider to access only one
4 page and that page is a "spider-friendly main" Web page (i.e., "spider-friendly
5 index" page). The spider-friendly main page is generated upon access. The
6 generation of the spider-friendly main page is illustrated in Fig. 4, which is
7 discussed below.

8 Therefore, if the static address parsed from the request is an "alias address",
9 then the process will proceed to block 208 for further analysis of the static address.
10 Otherwise, the static address is a conventional static address and it points to an
11 actual static Web page. In this case, the process will proceed to block 206 to allow
12 access to an actual static Web page as normal. The filter will hand the static
13 address to the Web server so that the Web server can access and send the
14 referenced static Web page.

15 At 208 of Fig. 3, the static address (which is now known to be an alias
16 address) is further examined to determine if the address includes parameter values.
17 Parameter values are fields after the path in the static address. Referring again to
18 Fig. 2, "/valx/vallast.htm" are parameter values after "path" in static address 172
19 of block 170.

20 If the static address does not have parameter values, then the static address
21 points to the spider-friendly main page. Therefore, the process proceeds to block
22 210 and the generation of the spider-friendly main page illustrated in Fig. 4, which
23 is discussed below. If the static address includes parameter values, then the static
24 address points to dynamic Web page, then the process proceeds to block 212.
25

1 At 212, the static address is converted into its mapped dynamic address in
2 the manner illustrated from bottom to top in Fig. 2 and described above as the S-
3 to-D address conversion. This produces a dynamic address that points to an
4 existing files storing instructions from dynamically generating the dynamic Web
5 page.

6 At 214, the dynamic Web page referenced by the converted dynamic
7 address is invoked. When generating the web page, meta-tags may also be
8 inserted. Such meta-tags help the search engines properly index each individual
9 web page. The meta-tags may be derived from a variety of sources. For example,
10 the meta-tags may be dynamically generated from the content of the web page.
11 Alternatively, the meta-tags may be retrieved from the database. In yet a further
12 alternative, the meta-tags may be from a variety of sources, such as a combination
13 of meta-tags that are dynamically generated and meta-tags that are retrieved from
14 a database or provided by some other component.

15 The purpose of inserting these meta-tags is to enhance the chances of the
16 specific page being found during a search of a search engine. The use of meta-
17 tags increases the breadth of keywords used on the page. This increases the
18 likelihood of the specific page being found during a search on a search engine. At
19 216, the dynamic Web page is sent to the requester. Dynamic Web pages have
20 instructions and the dynamic addresses have parameters and values. These
21 instructions, the parameters, the values are used to generate (i.e., invoke) a Web
22 page based upon information in the database. At 218, on-going Web page access
23 continues as normal.
24
25

1 In blocks 204 and 208, the filter makes three determinations in this order:

- 2 • Does the request include a normal static address to access actual
3 static Web pages?
- 4 • If not, then does the request include a “special” static address? A
5 static address is “special” if it includes an alias-indicator flag.
- 6 • If the static address is “special”, then does the address point to the
7 spider-friendly main page or does it point to somewhere else?

8 The exemplary converter makes these three determinations by detecting an alias-
9 indicator flag (like “flag” of block 162 of Fig. 2) and parameter values (like “valx”
10 and “vallast” of blocks 148 and 150 of Fig. 2).

11 Alternatively, the determinations may be made by employing multiple flags
12 in the static address. Each flag helps the filter make each determination. In
13 addition, the spider-friendly main page may have a defined address that contains
14 no flags itself, but the filter will recognize the defined address.

15 Fig. 4 shows more of the process for handling a request for dynamic Web
16 pages that is implemented by the Web address converter. Specifically, Fig. 4
17 shows the generation of the spider-friendly main page, which is represented by
18 block 210 of Fig. 3.

19 At 250, the filter dynamically generates an initial main page. This initial
20 main page includes links containing dynamic addresses pointing to dynamic Web
21 pages. These dynamic Web pages may represent the current inventory for an e-
22 commerce site.

23 At 252, meta-tags are inserted into the initial main page. The filter has a
24 defined set of meta-tags that are inserted each time a main page is generated.
25 Alternatively, the filter dynamically generates a set of meta-tags that are created

1 based upon current information in the database. Further, the filter may directly
2 retrieve the meta-tags from the database.

3 Like the meta-tags generated for a specific dynamic Web page of block
4 214, the purpose of inserting these meta-tags into the main page is to enhance the
5 chances of the page being found during a search of a search engine. The use of
6 meta-tags increases the breadth of keywords used on the page. This increases the
7 likelihood of the main page being found during a search on a search engine.

8 At 254, the dynamic addresses in these links are converted into static
9 addresses in the manner described above as D-to-S address conversion and shown
10 from top to bottom in Fig. 2.

11 At 256, the filter sends the spider-friendly main page to the requester. The
12 requester is typically a browser or a spider. At 258, on-going Web page access
13 continues as normal.

14 15 Exemplary Computer

16 Fig. 5 shows a computer that is an example of a Web server that is used
17 with the exemplary implementation of the Web address converter for dynamic
18 Web pages. This computer may be a client running a Web browser, a Web server
19 running a search engine, a Web server hosting a static Web site, a Web server
20 hosting a dynamic Web site, or any computer within a communications network
21 (such as the Internet).

22 As shown in Fig. 5, computer 330 includes one or more processors or
23 processing units 332, a system memory 334, and a bus 336 that couples various
24 system components including the system memory 334 to processors 332. Bus 336
25 represents one or more of any of several types of bus structures, including a

memory bus or memory controller, a peripheral bus, an accelerated graphics port, and a processor or local bus using any of a variety of bus architectures.

The system memory includes read only memory (ROM) 338 and random access memory (RAM) 340. A basic input/output system (BIOS) 342, containing the basic routines that help to transfer information between elements within computer 330, such as during start-up, is stored in ROM 338.

Computer 330 further includes a hard disk drive 344 for reading from and writing to a hard disk, not shown, a magnetic disk drive 346 for reading from and writing to a removable magnetic disk 348, and an optical disk drive 350 for reading from or writing to a removable optical disk 352 such as a CD ROM, DVD ROM or other optical media. The hard disk drive 344, magnetic disk drive 346 and optical disk drive 350 are each connected to bus 336 by one or more interfaces 354.

The drives and their associated computer-readable media provide nonvolatile storage of computer readable instructions, data structures, program modules, and other data for computer 330. Although the exemplary environment described herein employs a hard disk, a removable magnetic disk 348 and a removable optical disk 352, it should be appreciated by those skilled in the art that other types of computer readable media which can store data that is accessible by a computer, such as magnetic cassettes, flash memory cards, digital video disks, random access memories (RAMs), read only memories (ROM), and the like, may also be used in the exemplary operating environment.

A number of program modules may be stored on the hard disk, magnetic disk 348, optical disk 352, ROM 338, or RAM 340, including an operating system 358, one or more application programs 360 (such as a Web browser), other

1 program modules 362, and program data 364. A user may enter commands and
2 information into computer 330 through input devices such as keyboard 366 and
3 pointing device 368. Other input devices (not shown) may include a microphone,
4 joystick, game pad, satellite dish, scanner, or the like. These and other input
5 devices are connected to the processing unit 332 through an interface 370 that is
6 coupled to bus 336.

7 A monitor 372 or other type of display device is also connected to bus 336
8 via an interface, such as a video adapter 374. In addition to the monitor, personal
9 computers typically include other peripheral output devices (not shown) such as
10 speakers and printers.

11 Computer 330 can operate in a networked environment using logical
12 connections to one or more remote computers, such as a Web server 382. Web
13 server 382 typically includes many or all of the elements described above relative
14 to computer 330. In addition, a Web database 384 may be connected to the Web
15 server 382.

16 A logical connection that is not depicted in Fig. 5 is a local area network
17 (LAN) via network interface 386 and a general wide area network (WAN) via a
18 modem 378. Such networking environments are commonplace in offices,
19 enterprise-wide computer networks, intranets, and the Internet.

20 Depicted in Fig. 5, is a specific implementation of a WAN via the Internet.
21 Over the Internet, computer 330 typically includes a modem 378 or other means
22 for establishing communications over the Internet 380. Modem 378, which may
23 be internal or external, is connected to bus 336 via interface 356.

24 In a networked environment, program modules depicted relative to the
25 personal computer 330, or portions thereof, may be stored in the remote memory

1 storage device. It will be appreciated that the network connections shown and
2 described are exemplary and other means of establishing a communications link
3 between the computers may be used.

4
5 Example: "Acmetoys.store" Dynamic Web Site

6 The following example is provided to help illustrate how the exemplary
7 implementation of the Web address converter might be used.

8 Suppose that a fictional company called "Acme Toy Store" has an e-
9 commerce Web site where it sells toys and playthings. Acme's Web address is
10 "http://acmetoys.store". Like a "brick-and-mortar" store, Acme's e-commerce site
11 has an inventory that is constantly changing as existing items are sold and shipped
12 and new items arrive. Therefore, Acme's Web site has dynamically accessible and
13 updateable database for tracking inventory. Acme can track its inventory in real
14 time and users of Acme's site can order "in-stock" items in real time.

15 Acme would like for Internet search engines to be a source of inexpensive
16 advertisement for its toys. Acme would like for the spiders of search engines to
17 crawl its Web site and index its products. That way, a search engine may direct a
18 user (who is searching for that hot new toy from overseas) to Acme's site.
19 Preferably, the search engine may direct the user to the particular page on the site
20 referring to that hot new toy. Acme would like users to go directly to their
21 dynamic Web pages so that the user can view the current product information.

22 Acme implements the exemplary embodiment of the Web address converter
23 in an ISAPI filter on its Web site. Acme modifies its "robots.txt" file on its Web
24 server to authorize and direct spiders to crawl a spider-friendly main page at
25 "http://acmetoys.store/flag/index.htm". The alias-indicator flag is "flag."

1 Subsequently, a spider from a search engine arrives to crawl Acme's site.
2 The spider examines the "robots.txt" file and proceeds to access the spider-
3 friendly main page.

4 Referring to the flow charts shown in Figs. 3 and 4, the filter receives (at
5 200 of Fig. 3) a request from the spider to access a Web page. The filter parses (at
6 202) the request and pulls out this static address:
7 "http://acmetoys.store/flag/index.htm".

8 Upon examination (at 204) of this address, the filter determines that it
9 includes "flag", which is the alias-indicator flag. Upon further examination (at
10 208) of this address, the filter determines that the address does not include any
11 parameter values.

12 Therefore, the filter generates an initial main page having links with
13 dynamic addresses (at 250 of Fig. 4). The dynamic addresses point to dynamic
14 Web pages describing the available toys and providing a means for purchasing
15 such toys. The filter inserts (at 252) a set of meta-tags into the initial main page.
16 The meta-tags have many keywords related to toys and playthings.

17 D-to-S conversion (at 254) is performed on the dynamic address so that the
18 links include static addresses. These static addresses will include the alias-
19 indicator flag and one or more parameter values.

Below is an example of Web address conversion using a dynamic Web page on Acme's site that describes a fictional product called "megawheel." The following example of D-to-S address conversion is done in accordance with such D-to-S address conversion illustrated in Fig. 2 and described above:

Megawheel dynamic address:

`http://acmetoys.store/toys/wheeled.asp?age=4-or-under&name=megawheel`

The following is a table identifying the fields illustrated in Fig. 2 and described above. The field names are italicized. The table also gives the specific values of these fields:

Common Fields		Dynamic Fields Removed		Static Fields Added	
<i>scheme</i>	http://	<i>field separator</i>	/	<i>field separator</i>	/
<i>hostname</i>	acmetoys.store	<i>dynamic file extension</i>	".asp"	<i>flag</i>	flag
<i>path</i>	toys/wheeled	<i>file-parameter-separator</i>	?	<i>field separator</i>	/
<i>valx</i>	4-or-under	<i>parameter</i>	age	<i>field separator</i>	/
<i>vallast</i>	megawheel	<i>parameter separator</i>	&	<i>field separator</i>	/
		<i>last parameter</i>	name	<i>static file extension</i>	".htm"

Resulting example static address (after D-to-S address conversion):

Megawheel static address:

`http://acmetoys.store/flag/toys/wheeled/4-or-under/megawheel.htm`

1 After the D-to-S address conversion of the address in the links in the initial
2 main page, the page becomes the spider-friendly main page. The filter sends (at
3 256) the spider-friendly main page to the original requester, which was the spider
4 of the search engine.

5 The indexer of the search engine indexes the spider-friendly main page of
6 Acme's Web site. Keywords from the spider-friendly main page are stored in the
7 search engine's database.

8 Subsequently, a user searches for a toy called "megawheel" on the search
9 engine. It discovers a link to Acme's Web site. That link includes the megawheel
10 static address (as shown above). The user clicks on that link and is whisked away
11 to Acme's site.

12 Again referring to the flow charts shown in Figs. 3 and 4, the filter receives
13 (at 200 of Fig. 3) a request from the browser of the user to access a Web page.
14 The filter parses (at 202) the request and pulls out the megawheel static address:
15 "http://acmetoys.store/flag/toys/wheeled/4-or-under/megawheel.htm".

16 Upon examination (at 204) of this address, the filter determines that it
17 includes "flag", which is the alias-indicator flag. Upon further examination (at
18 208) of this address, the filter determines that the address includes parameter
19 values. Specifically, the address includes "4-or-under" and "megawheel"
20 parameter values. The filter performs S-to-D address conversion (at 212) on the
21 static address so that browser can be redirected to the megawheel dynamic Web
22 page.
23
24
25

1 S-to-D address conversion is done in accordance with such S-to-D
2 conversion illustrated in Fig. 2 and described above. The static address is
3 converted into megawheel dynamic address:
4 “http://acmetoys.store/toys/wheeled.asp?age=4-or-under&name=megawheel”.

5 Note that this is the megawheel dynamic address and the same address that was
6 the source for the D-to-S address conversion performed when generating the
7 spider-friendly main page.

8 After this S-to-D address conversion, the desired megawheel dynamic Web
9 page is invoked and megawheel-related meta tags are inserted into the page (at
10 214). The page is sent to the browser of the user (at 216). Therefore, the user
11 views the current dynamic Web page for the “megawheel” product. The user found
12 the megawheel dynamic Web page using a corresponding megawheel static
13 address stored in a searchable database of a search engine.

14 15 Conclusion

16 Using the exemplary implementation of the Web address converter, spiders
17 can fetch and indexers can index the dynamic content of dynamic Web sites.
18 Furthermore, browsers using a static address to a dynamic Web page of a dynamic
19 Web site can access the referenced dynamic Web page, rather than a stale, static
20 copy.

21 Although the address converter has been described in language specific to
22 structural features and/or methodological steps, it is to be understood that the web
23 address converter defined in the appended claims is not necessarily limited to the
24 specific features or steps described. Rather, the specific features and steps are
25 disclosed as preferred forms of implementing the claimed web address converter.

CLAIMS

1. A spider-friendly Web page generation method comprising:
generating an instance of a main Web page having at least one link with a
dynamic address pointing to a dynamic Web page; and
converting the dynamic address into a static address that also points to the
dynamic Web page.

2. A method as recited in claim 1 further comprising receiving a request
for an instance of the main Web page before the generating.

3. A method as recited in claim 1 further comprising sending the
instance of the main Web page.

4. A method as recited in claim 1 further comprising receiving a request
for access to the main Web page, the request comprising a static address pointing
to the main Web page.

5. A method as recited in claim 1, wherein the generating comprises
forming the instance of the main Web page so that the main Web page contains
meta-tags for facilitating indexing by a Web search engine.

1 6. A method as recited in claim 1, wherein the converting comprises:
2 parsing the dynamic address to identify and separate fields within the
3 dynamic address, wherein at least one field has a value; and
4 generating a static address incorporating the value of at least one field,
5 wherein the static address points to the dynamic Web page.

6
7 7. A computer-readable storage medium having computer-executable
8 instructions that, when executed by a computer, performs the method as recited in
9 claim 1.

10
11 8. A static to dynamic (S-to-D) Web address conversion method
12 comprising:

13 receiving a request for a dynamic Web page, the request including a static
14 address pointing to the dynamic Web page; and

15 converting the static address to a dynamic address also pointing to the
16 dynamic Web page.

17
18 9. A method as recited in claim 8 further comprising providing the
19 dynamic address to a server.

20
21 10. A method as recited in claim 8 further comprising invoking the
22 dynamic Web page referenced by the dynamic address.

1 **11.** A method as recited in claim 8 further comprising sending the
2 dynamic Web page referenced by the dynamic address to a requester.

3
4 **12.** A method as recited in claim 8, wherein the converting comprises:
5 parsing the static address to identify at least one value associated with a
6 field within the static address; and
7 generating a dynamic address incorporating at least one value associated
8 with a field, wherein the dynamic address points to the dynamic Web page.

9
10 **13.** A computer-readable storage medium having computer-executable
11 instructions that, when executed by a computer, performs the method as recited in
12 claim 8.

13
14 **14.** A static address to dynamic address (S-to-D address) conversion
15 method comprising:

16 receiving a static address pointing to a dynamic Web page;
17 parsing the static address to identify at least one value associated with a
18 field within the static address; and
19 generating a dynamic address incorporating at least one value associated
20 with a field, wherein the dynamic address points to the dynamic Web page.

1 **15.** A method as recited in claim 14, wherein the parsing identifies and
2 separates fields comprising:

3 a “hostname” for containing a value that a name of a server hosting the
4 dynamic Web page;

5 an “alias-indicator” for containing a value that identifies the static address
6 as pointing to a dynamic Web page rather than a static Web page;

7 a “path” for containing a value that a hierarchical path name of the dynamic
8 Web page; and

9 “searchvalues” for containing one or more values, wherein each value is
10 associated with a name of parameters used to invoke the dynamic Web page.

11
12 **16.** A method as recited in claim 15, wherein the generating forms the
13 dynamic address by incorporating values of at least the following fields:

14 the hostname;

15 the path; and

16 the searchvalues.

17
18 **17.** A method as recited in claim 16, wherein the generating forms the
19 dynamic address by incorporating a file-parameter-separator symbol.

20
21 **18.** A method as recited in claim 16, wherein the generating forms the
22 dynamic address by incorporating a “?” symbol.

1 **19.** A method as recited in claim 15, wherein the generating forms the
2 dynamic address by incorporating, at least and in order relative to each other, the
3 following:

4 the value of the hostname;

5 the value of the path;

6 a file-parameter-separator symbol;

7 one or more values of a "searchparameters" field, wherein the
8 searchparameters contain one or more values that are names of parameters used to
9 invoke the dynamic Web page; and

10 a value for each searchvalue, wherein the value of each searchvalue is
11 associated with a name of the parameters of the searchparameters.

12
13 **20.** A method as recited in claim 19, wherein each value of a field is
14 separated by a field-separator symbol.

15
16 **21.** A method as recited in claim 19, wherein the path is followed by a
17 format-identifying extension.

18
19 **22.** A computer-readable storage medium having computer-executable
20 instructions that, when executed by a computer, performs the method as recited in
21 claim 14.

1 **23.** A dynamic to static (D-to-S) Web address conversion method
2 comprising:

3 receiving a dynamic address pointing to a dynamic Web page; and

4 converting the dynamic address to a static address also pointing to the
5 dynamic Web page.

6
7 **24.** A computer-readable storage medium having computer-executable
8 instructions that, when executed by a computer, performs the method as recited in
9 claim 23.

10
11 **25.** A dynamic to static (D-to-S) Web address conversion method
12 comprising:

13 receiving a dynamic address pointing to a dynamic Web page;

14 parsing the dynamic address to identify and separate fields within the
15 dynamic address, wherein at least one field has a value; and

16 generating a static address incorporating the value of at least one field, the
17 static address also pointing to the dynamic Web page.

1 **26.** A method as recited in claim 25, wherein the parsing identifies and
2 separates fields comprising:

3 a “hostname” for containing a value that is a name of a server hosting the
4 dynamic Web page;

5 a “path” for containing a value that a hierarchical path name of the dynamic
6 Web page;

7 “searchparameters” for containing one or more values that are names of
8 parameters used to invoke the dynamic Web page; and

9 “searchvalues” for containing one or more values, wherein each value is
10 associated with a name of a parameter in the searchparameters field.

11
12 **27.** A method as recited in claim 26, wherein the generating forms the
13 static address by incorporating the values of at least the following fields:

14 the hostname;

15 the path; and

16 the searchvalues.

17
18 **28.** A method as recited in claim 27, wherein the generating forms the
19 static address by incorporating a value for an alias-indicator field identifying the
20 static address as pointing to a dynamically generated Web page.

1 **29.** A method as recited in claim 28, wherein the generating forms the
2 static address by incorporating, at least and in order relative to each other, the
3 following:

4 the value of the hostname;
5 the value for an alias-indicator field;
6 the value of the path; and
7 the value of each searchvalue.

8
9 **30.** A method as recited in claim 29, wherein each value of a field being
10 separated by a field-separator symbol.

11
12 **31.** A method as recited in claim 26, wherein the generating step forms
13 the static address wherein last of the searchvalues is followed by a format-
14 identifying extension.

15
16 **32.** A method as recited in claim 31, wherein the format-identifying
17 extension is either “.htm” or “.html”.

18
19 **33.** A computer-readable storage medium having computer-executable
20 instructions that, when executed by a computer, performs the method as recited in
21 claim 25.

1 **34.** A method for facilitating the indexing of a collection of dynamic
2 web pages, the method comprising:

3 storing a file authorizing a web spider to access a spider-friendly Web page
4 having a designated static address;

5 upon attempted access by a Web spider of the spider-friendly Web page,
6 generating an instance of the spider-friendly Web page by:

7 identifying a plurality of dynamic Web pages that are to be indexed
8 by the Web spider, each of the dynamic Web pages including at least one
9 parameter value that is used to retrieve dynamic content to be included in
10 the Web page;

11 for each of the dynamic Web pages, creating a static Web address
12 that identifies the dynamic Web page, the static address encoding a
13 parameter value used to retrieve dynamic content for the Web page;

14 assembling the static addresses corresponding to the identified
15 dynamic Web pages into an instance of the spider-friendly Web page; and

16 sending the spider-friendly Web page to the Web spider.

17
18 **35.** A method of providing a dynamic Web page comprising:

19 receiving a request for a dynamic Web page from a computer on a network,
20 the request including a static Web address pointing to the dynamic Web page;

21 generating an instance of the dynamic Web page such that contents of the
22 instance appears as a static Web page; and

23 sending the dynamic Web page to a the computer.
24
25

1 **36.** A data structure for use with a computer having a processor and a
2 memory, the data structure encoding a static Web address that can be indexed by
3 Web spider and that points to a dynamic Web page comprising, said structure
4 comprising:

5 a first portion indicating a host name where the dynamic Web page resides;
6 a second portion encoding a path where the dynamic Web page resides; and
7 a third portion encoding at least one parameter value that may be used to
8 generate the dynamic Web page, the third portion being encoded such that a Web
9 spider will identify the static address as a valid static address.

10
11 **37.** A method as recited in claim 36 further comprising a fourth portion
12 encoding a token recognized by a Static-to-Dynamic (S-to-D) Web address
13 converter.

14
15 **38.** A method as recited in claim 36, wherein the third portion encodes
16 at least one parameter associated with the one value, the parameter also may be
17 used to generate the dynamic Web page

18
19 **39.** A data structure for use with a computer having a processor and a
20 memory, said structure comprising a static Web address pointing to a dynamic
21 Web page stored on the computer.

1 **40.** A data structure for use with a first computer having a processor and
2 a memory, said structure comprising a static Web address pointing to a dynamic
3 Web page, wherein the dynamic Web page is stored on a second computer having
4 a processor and a memory, the first and second computers being operatively
5 coupled via a communications network.

6
7 **41.** A Web site system comprising:
8 a Web server hosting a dynamic Web site;
9 a database storing data used by the Web server to generate dynamic Web
10 pages of the dynamic Web site, the Web server being operatively coupled to the
11 Web server; and
12 a static to dynamic (S-to-D) Web address converter, the converter being
13 operatively coupled to the Web server;
14 the S-to-D Web address converter being configured to convert a static
15 address to a dynamic address pointing to a dynamic Web page.

16
17
18
19
20
21
22
23
24
25

1 **42.** A Web site system comprising:

2 a Web server hosting a dynamic Web site;

3 a database storing data used by the Web server to generate dynamic Web
4 pages of the dynamic Web site, the Web server being operatively coupled to the
5 Web server; and

6 a dynamic to static (D-to-S) Web address converter, the converter being
7 operatively coupled to the Web server;

8 the D-to-S Web address converter being configured to convert a dynamic
9 address pointing to a dynamic Web page into a static address also pointing to the
10 dynamic Web page.

11
12 **43.** A server comprising:

13 a processor;

14 a request receiver executable on the processor to receive a request including
15 a static address of a main Web page;

16 a spider-friendly Web page generator executable on the processor to:

17 receive the static address of the main Web page from the request
18 receiver;

19 generate an instance of the main Web page having at least one link
20 with an address pointing to a dynamic Web page.

1 **44.** A server comprising:

2 a processor;

3 a static to dynamic (S-to-D) Web address converter executable on the
4 processor to:

5 convert a static address pointing to a dynamic Web page into a
6 dynamic address that also points to the dynamic Web page.

7
8 **45.** A server comprising:

9 a processor;

10 a static to dynamic (S-to-D) Web address converter executable on the
11 processor to:

12 parse the static address to identify at least one value associated with
13 a field within the static address; and

14 generating a dynamic address incorporating at least one value
15 associated with a field, wherein the dynamic address points to the dynamic.
16 Web page.

17
18 **46.** A server comprising:

19 a processor;

20 a dynamic to static (D-to-S) Web address converter executable on the
21 processor to:

22 convert a dynamic address pointing to a dynamic Web page into a
23 static address also pointing to the dynamic Web page.

1
2
3 47. A system for hosting dynamic Web sites comprising:
4 a Web server for dynamically generating an instance of a dynamic Web
5 page; and
6 a spider-friendly Web page generator configured to:
7 generate an instance of a main Web page having at least one link
8 with a dynamic address pointing to a dynamic Web page; and
9 convert the dynamic address into a static address that also points to
10 the dynamic Web page.
11
12
13
14
15
16
17
18
19
20
21
22
23
24
25

1 **48.** A system for hosting dynamic Web sites comprising:
2 a Web server for dynamically generating an instance of a dynamic Web
3 page in response to a request; and
4 a static to dynamic (S-to-D) Web address converter;
5 the Web server being configured to send a Web address of the request to
6 the converter;
7 the converter being configured to:
8 receive the Web address of the request;
9 determine if the Web address is a static address; and
10 convert the static address to a dynamic address that also points to the
11 dynamic Web page.

12
13 **49.** A system recited in claim 48, the converter being further configured
14 to convert the static address to a dynamic address by:
15 parsing the static address to identify at least one value associated with a
16 field within the static address; and
17 generating a dynamic address incorporating at least one value associated
18 with a field, wherein the dynamic address points to the dynamic Web page.

1 **50.** A system for hosting dynamic Web sites for hosting a dynamic Web
2 site comprising:

3 a Web server for dynamically generating an instance of a dynamic Web
4 page; and

5 a dynamic-to-static (D-to-S) Web address converter being configured to:

6 convert a dynamic address pointing to a dynamic Web page into a
7 static address that also points to the dynamic Web page.

8
9 **51.** The system recited in claim 50, the converter being further
10 configured to convert the dynamic address to a static address by:

11 parsing the dynamic address to identify and separate fields within the
12 dynamic address, wherein at least one field has a value; and

13 generating a static address incorporating the value of at least one field, the
14 static address also pointing to the dynamic Web page.

15
16 **52.** A computer-readable storage medium having computer-executable
17 instructions that, when executed by a computer, performs the spider-friendly Web
18 page generation method comprising:

19 generating an instance of a spider-friendly Web page having at least one
20 link with a dynamic address pointing to a dynamic Web page; and

21 converting the dynamic address into a static address that also points to the
22 dynamic Web page.

23
24
25

1 **53.** A computer-readable storage medium having computer-executable
2 instructions that, when executed by a computer, performs the static to dynamic (S-
3 to-D) Web address conversion method comprising:

4 receiving a request for a dynamic Web page, wherein the request includes a
5 static address pointing to the dynamic Web page; and

6 converting the static address to a dynamic address that also points to the
7 dynamic Web page.

8
9 **54.** A computer-readable storage medium having computer-executable
10 instructions that, when executed by a computer, performs the static to dynamic (S-
11 to-D) Web address conversion method comprising:

12 receiving a static address pointing to a dynamic Web page;

13 parsing the static address to identify at least one value associated with a
14 field within the static address; and

15 generating a dynamic address incorporating at least one value associated
16 with a field, wherein the dynamic address points to the dynamic Web page.

17
18
19 **55.** A computer-readable storage medium having computer-executable
20 instructions that, when executed by a computer, performs the dynamic to static (D-
21 to-S) Web address conversion method comprising:

22 receiving a dynamic address pointing to a dynamic Web page; and

23 converting the dynamic address to a static address that also to the dynamic
24 Web page

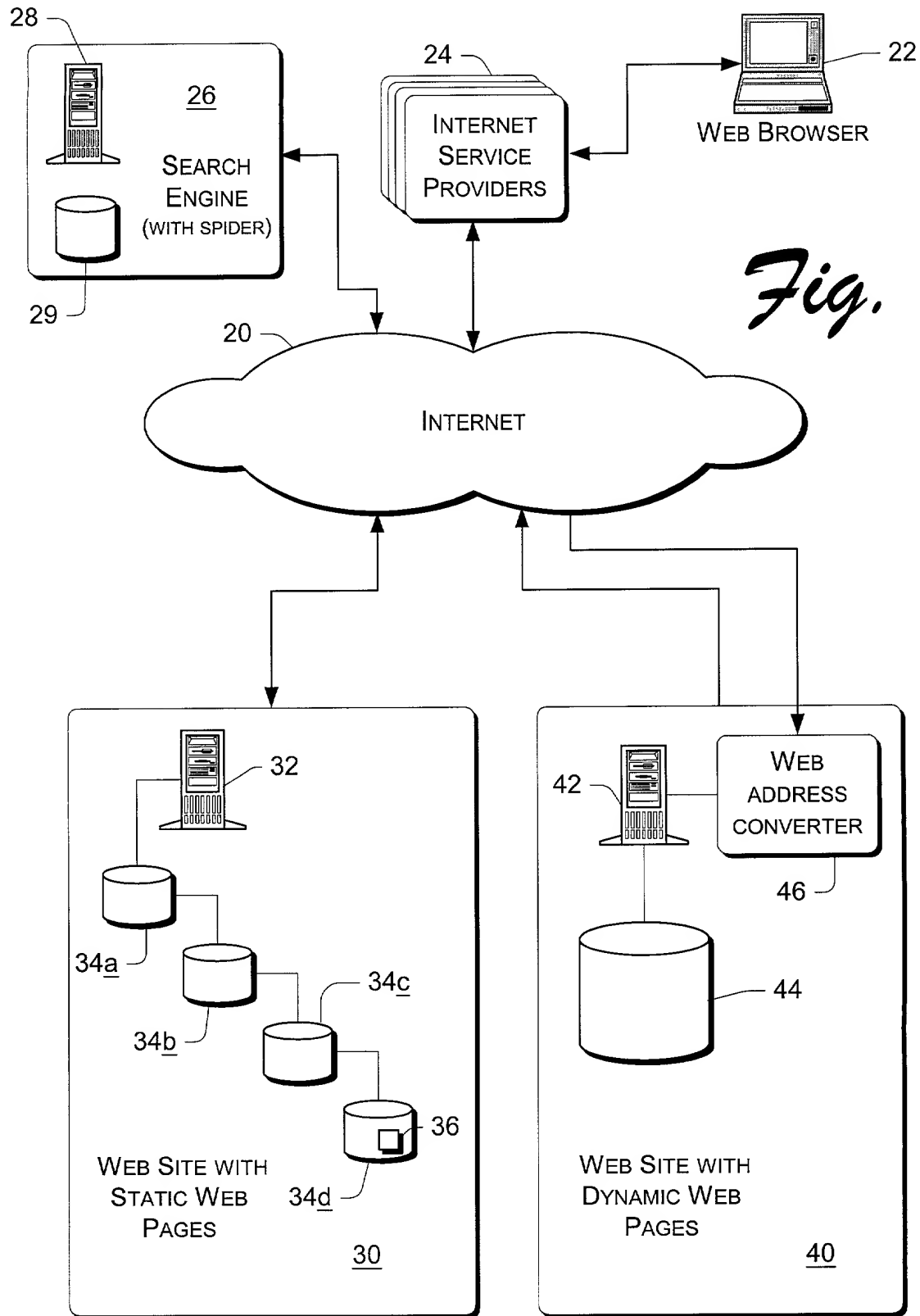
1 **56.** A computer-readable storage medium of claim 55 the instructions
2 for performing converting comprising:

3 parsing the dynamic address to identify and separate fields within the
4 dynamic address, wherein at least one field has a value; and

5 generating a static address incorporating the value of at least one field,
6 wherein the static address points to the dynamic Web page.
7
8
9
10
11
12
13
14
15
16
17
18
19
20
21
22
23
24
25

1 **ABSTRACT**

2 A Web address converter helps dynamic Web sites get the attention of
3 spiders of Internet search engines. With the Web address converter, requests from
4 Web browsers using static addresses access corresponding dynamic Web pages
5 and requests from search engines generate an instance of a Web page having links
6 with static addresses pointing to corresponding dynamic Web pages. The Web
7 address converter performs both Dynamic-to-Static (D-to-S) address conversion
8 and Static-to-Dynamic (S-to-D) address conversion. D-to-S address conversion is
9 done when generating a spider-friendly main page for a spider of a search engine
10 to crawl. S-to-D address conversion is used when a browser uses a static address
11 to access a corresponding dynamic Web page. The static address that the browser
12 uses was originally created when the spider-friendly main page was generated.
13
14
15
16
17
18
19
20
21
22
23
24
25



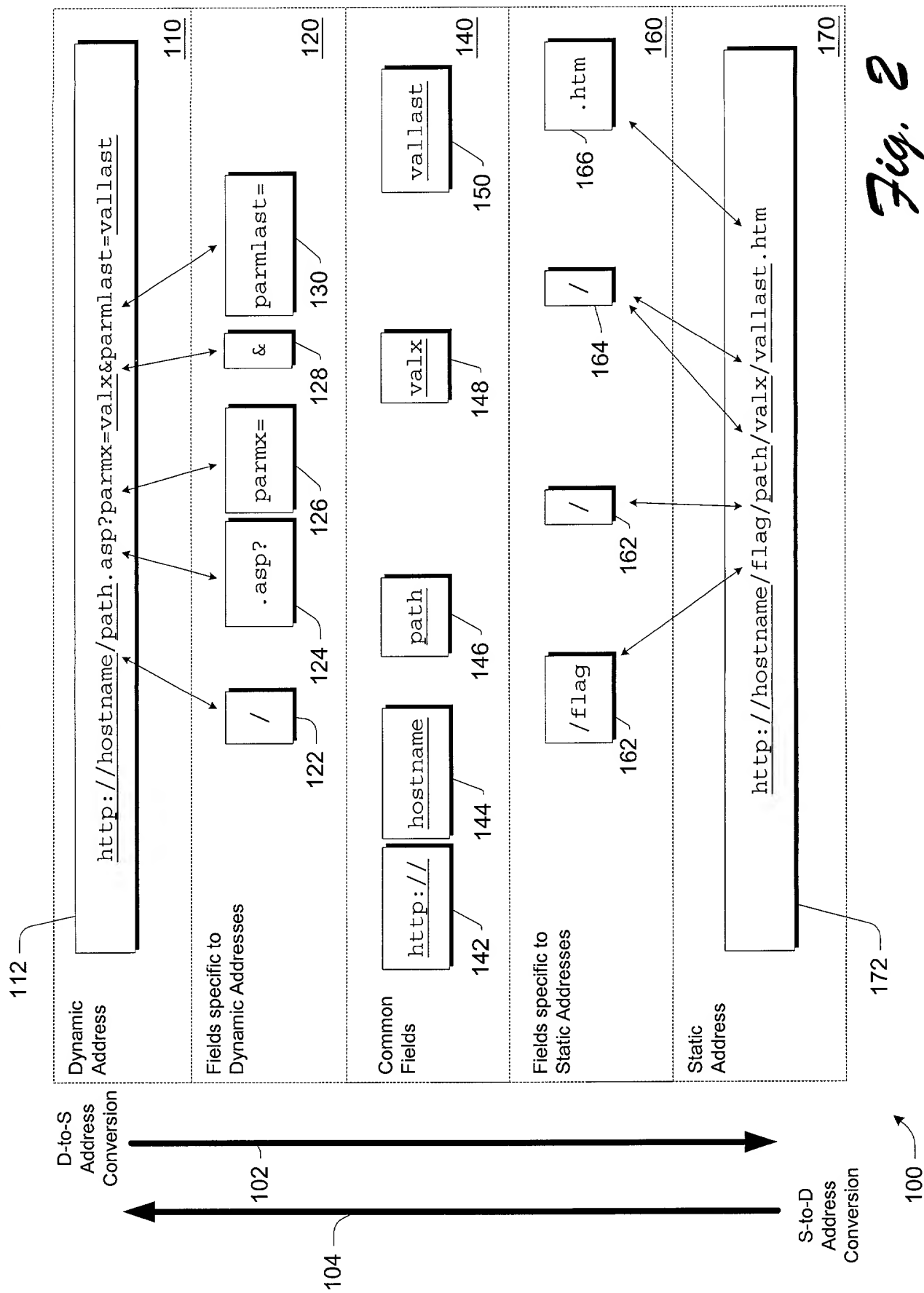
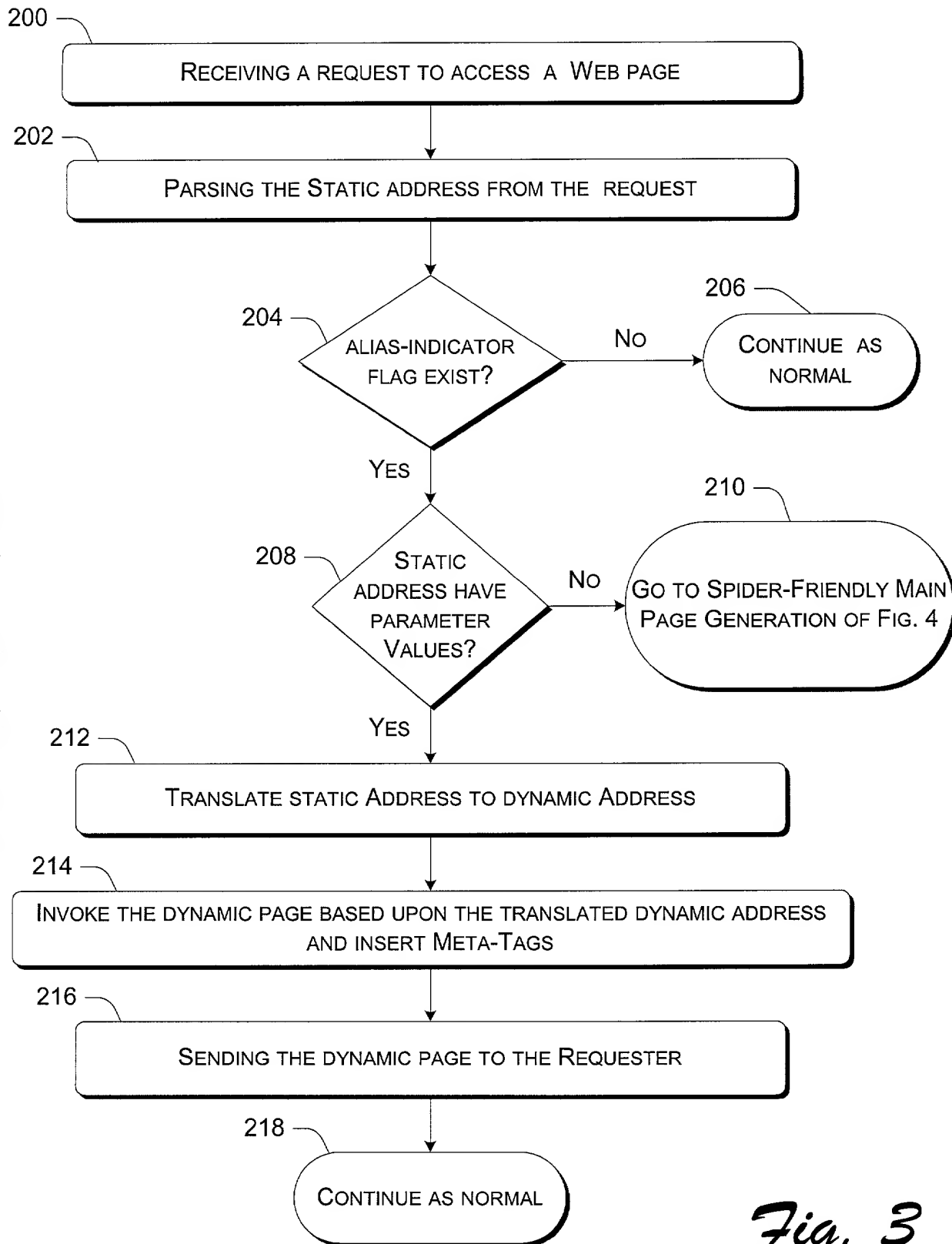


Fig. 2

*Fig. 3*

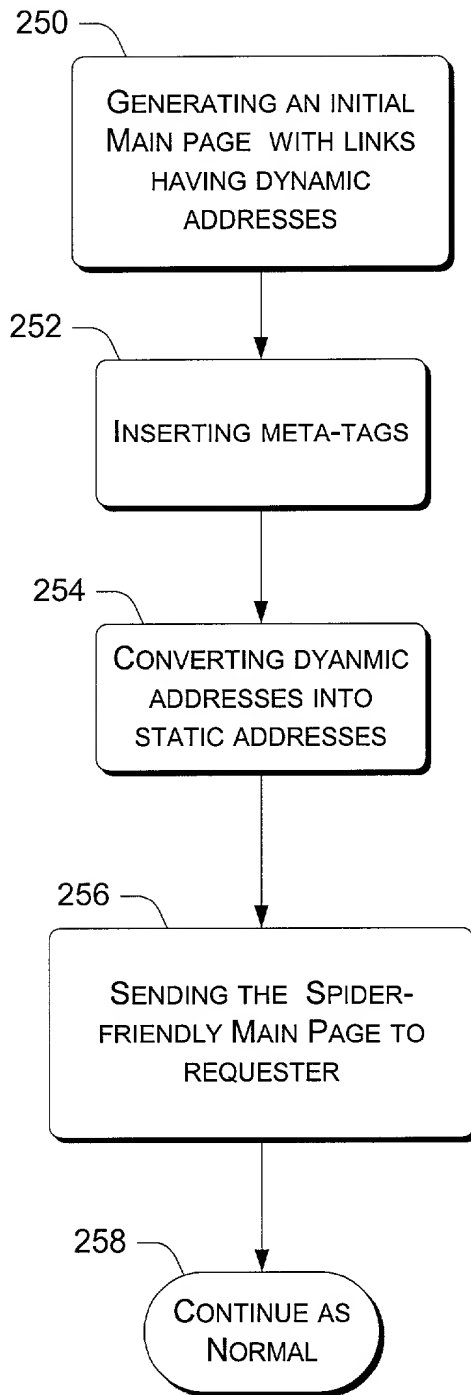
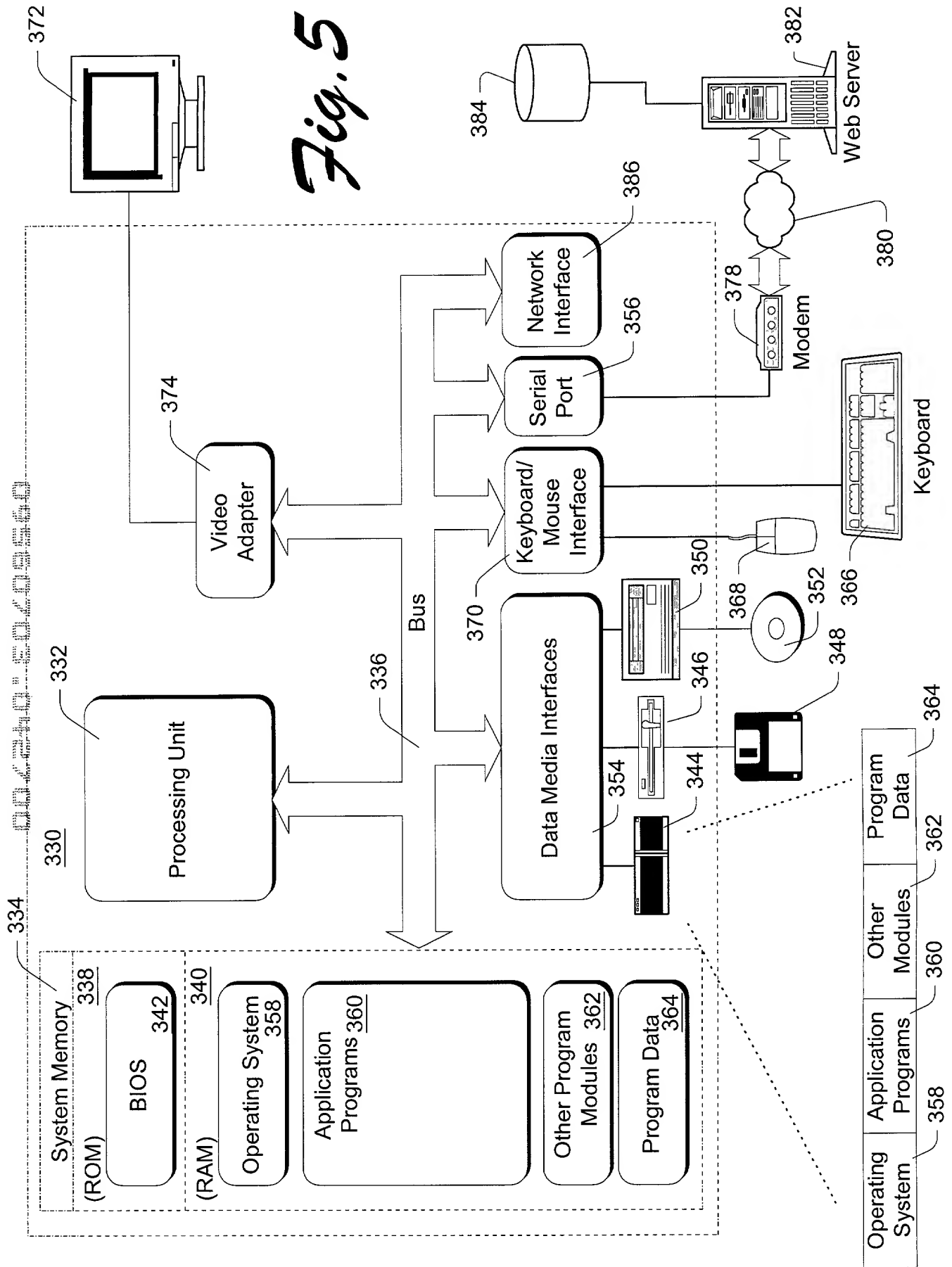


Fig. 4

Fig. 5



1 **IN THE UNITED STATES PATENT AND TRADEMARK OFFICE**

2 Inventorship..... Allen et al.
3 Applicant Microsoft Corporation
4 Attorney's Docket No. MS1-487US
5 Title: Web Address Converter for Dynamic Web Pages

6 **DECLARATION FOR PATENT APPLICATION**

7 As a below named inventor, I hereby declare that:

8 My residence, post office address and citizenship are as stated below next to
9 my name.

10 I believe I am the original, first and sole inventor (if only one name is listed
11 below) or an original, first and joint inventor (if plural names are listed below) of the
12 subject matter which is claimed and for which a patent is sought on the invention
13 entitled "Web Address Converter for Dynamic Web Pages," the specification of
14 which is attached hereto.

15 I have reviewed and understand the content of the above-identified
16 specification, including the claims.

17 I acknowledge the duty to disclose information which is material to the
18 examination of this application in accordance with Title 37, Code of Federal
19 Regulations, § 1.56(a).

20 PRIOR FOREIGN APPLICATIONS: no applications for foreign patents or
21 inventor's certificates have been filed prior to the date of execution of this
22 declaration.

23 **Power of Attorney**

24 I appoint the following attorneys to prosecute this application and transact all
25 future business in the Patent and Trademark Office connected with this application:
26 Lewis C. Lee, Reg. No. 34,656; Daniel L. Hayes, Reg. No. 34,618; Allan T.

1 Sponseller, Reg. 38,318; Steven R. Sponseller, Reg. No. 39,384; James R.
2 Banowsky, Reg. No. 37,773; Lance R. Sadler, Reg. No. 38,605; Michael A. Proksch,
3 Reg. No. 43,021; Thomas A. Jolly, Reg. No. 39,241; David A. Morasch, Reg. No.
4 42,905; Kasey C. Christie, Reg. No. 40,559; Katie E. Sako, Reg. No. 32,628 and
5 Daniel D. Crouse, Reg. No. 32,022.

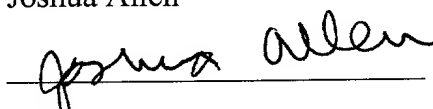
6 Send correspondence to: LEE & HAYES, PLLC, 421 W. Riverside Avenue,
7 Suite 500, Spokane, Washington, 99201. Direct telephone calls to: Kasey C.
8 Christie (509) 324-9256.

9
10 All statements made herein of my own knowledge are true and that all
11 statements made on information and belief are believed to be true; and further that
12 these statements were made with the knowledge that willful false statements and the
13 like so made are punishable by fine or imprisonment, or both, under Section 1001 of
14 Title 18 of the United States Code and that such willful false statement may
15 jeopardize the validity of the application or any patent issued therefrom.

16
17 * * * * *

18 Full name of inventor: Joshua Allen

19 Inventor's Signature

 Date: 4-26-2000

20 Residence: Kirkland, WA

21 Citizenship: USA

22 Post Office Address: 13130 NE 120th Lane B300
Kirkland, WA 98034

Full name of inventor:

Jeffrey L. Ranck

Inventor's Signature

Jeffrey L. Ranck

Date: 4/26/00

Residence:

Smmamish, WA

gr

Sammamish

Citizenship:

USA

Post Office Address:

22642 NE 15th Place
Smmamish, WA 98053

002240" E0703950